

Fonaments de programació amb Scratch

25-2-2021

Índex

Objectius d'aquesta activitat

¿Què és Scratch?

¿Com funciona Scratch?

Programa d'exemple

Variables

Operadors

Sentències condicionals

Sentències de repetició

Conclusions

1.Objectius d'aquesta activitat

En aquest document es pretén fer una altra aproximació a conceptes bàsics de la programació orientada a objectes donat que tothom aprendreu almenys un d'aquests durant el vostre temps d'estudi a la ItAcademy. En concret:

Nom itinerari	Llenguatge POO principal
React/Angular/Vue	Javascript / Typescript
NodeJS	Javascript
Java	Java
PHP	PHP
Data Science	Python

Tots els llenguatges que esmentem a la taula superior tenen en comú una sèrie de conceptes que es van introduir al document “Conceptes bàsics de programació”. Serà precisament aquests conceptes els que treballarem en aquest document des de l'òptica d'una eina especialment pensada per aprendre a programa, **Scratch**.

Val a dir però, que, **NO CAL QUE US CAPFIQUEU A DOMINAR SCRATCH**, sinó que, senzillament, caldrà que seguiu el guió que planteja aquest document i, un cop fet, us sentiu lliures de jugar amb Scratch, amb l'única pretensió però, de continuar familiaritzant-vos amb:

- Variables
- Operadors
- Estructures condicionals
- Estructures repetitives
- Funcions

Que serà una part força important de el que, ja si, programareu al bloc de **fonaments de la programació**.

2. Què és Scratch?

Es tracta d'un **llenguatge de programació visual**, que ens permet donar un cop d'ull lúcid a principis de la programació orientada a objectes però, sense endinsar-nos en el desenvolupament del codi font com a tal.

Va ser creat el 2012 per la Lifelong Kindergarten Group del MIT Media Lab.

Es pot obtenir fàcilment des de el seu portal oficial o, ser utilitzat directament online, com de fet, ha estat realitzat aquest petit tutorial.

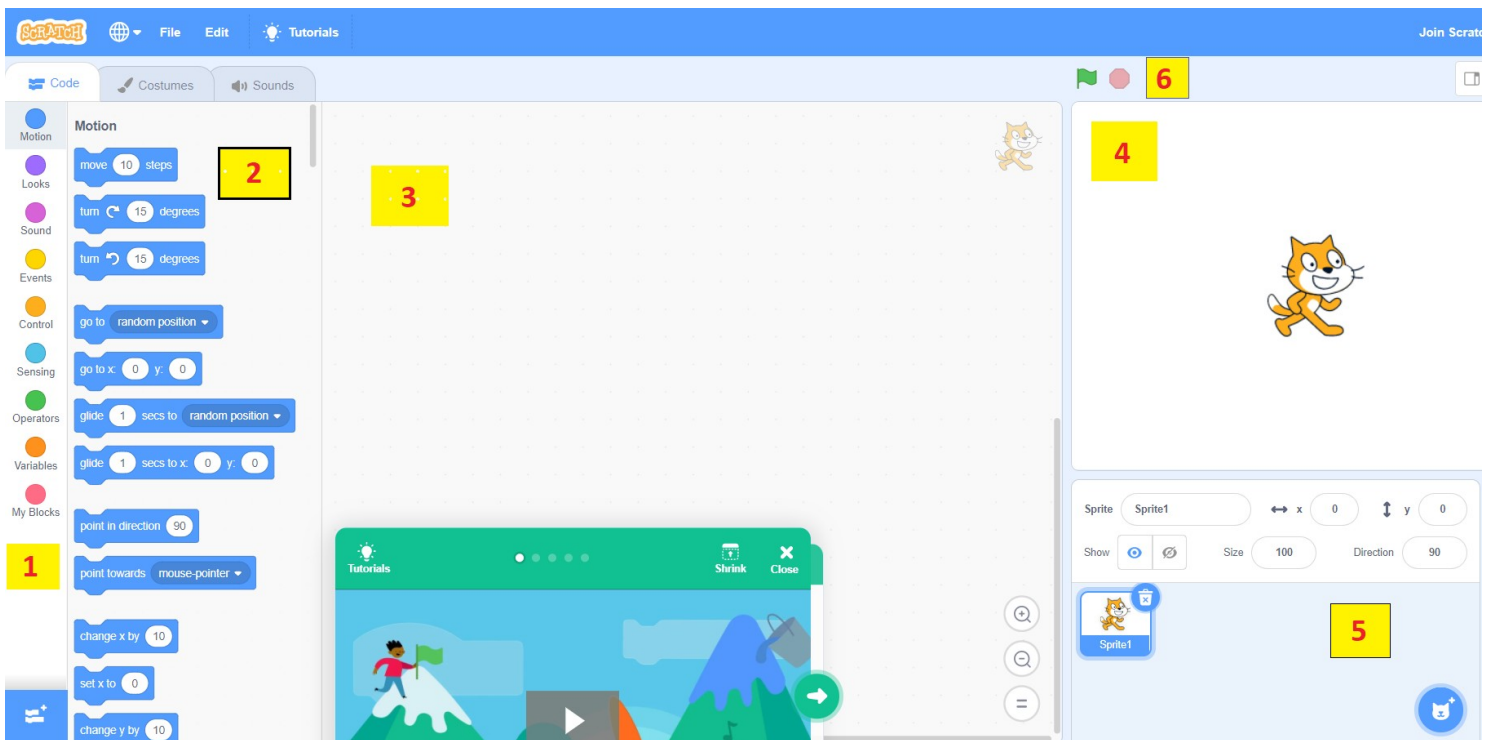
3. Com funciona Scratch?

Scratch funciona executant-se com un programa qualsevol, amb la principal diferència que tot el que es “programa” es mostra de forma molt visual, aïllant-nos una mica de l'abstracte que pot resultar programar en un principi.

Per fer-lo servir cal comprendre, doncs, algunes de les opcions que ens dóna.

Les instruccions de programació en Scratch venen representades visualment mitjançant **peces de trencaclosques**. Cada tipus d'instrucció (variable, operador, condicional...) ve representat per un tipus diferent de peça, i serà, mitjançant la correcta unió d'aquestes que crearem un programa.

Quant a les diferents parts de la vista, tindriem:



1- Llistat de tipus de peces: Segons si estem parlant de variables, operadors, o controladors de fluxe d'execució, veurem uns o altres tipus de peces, classificades en les seccions que aquí figuren. Nosaltres farem servir principalment **variables**, **operators**, **control** i alguna funció de la part de **motion** i **events**.

2- Peces: En funció de si triem una secció o altra, es veuran unes o altres peces en aquest apartat.

3- Zona de muntatge: Aquesta és la part predominant de la vista, on “muntarem” el nostre programa.

4- Zona d'output: Aquí hauríem de veure el resultat del nostre programa.

5- Selecció de *sprites*: La principal gràcia de Scratch és que el programa que muntem tindrà conseqüències visuals molt evidents, que ens ajudaran a comprendre els efectes reals de conceptes bàsics de la programació. Aquests efectes els podrem observar en els dibuixos que triem que apareguin en aquesta àrea, al nostre programa. Per defecte tenim el nostre estimat gat , però podem afegir més sprites en aquesta àrea de selecció. En principi no caldria tocar res d'aquesta àrea per aquesta pràctica, però.

6- Esdeveniments principals: En aquest apartat tenim:

- **Go:** La bandera verda, que utilitzarem per indicar que comença l'execució del programa.
- **Stop:** Per forçar la detenció d'un programa.

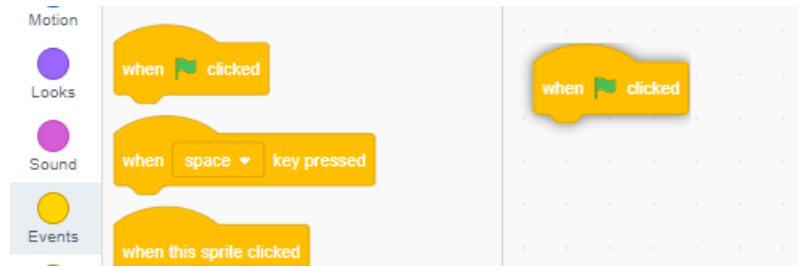
La bandera verda és el que fem servir per defecte per inicialitzar un programa. De tota manera, perquè això passi, hem d'indicar-ho al programa. Cosa que veurem en la següent secció.

4. Programa d'exemple

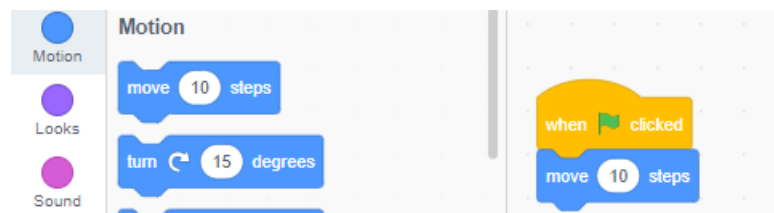
Imaginem que volem fer un programa per fer que el nostre estimat gat es desplaci:

- 1 - Primer cap a la dreta
- 2 - Després cap avall

Bé, com hem comentat just abans, per fer que un programa s'executi hem d'indicar que ho farà mitjançant el *click* a la bandera verda (tot i que també podem fer *click* en una única peça que estigui a l'àrea de muntatge per provar el seu efecte individual), la qual cosa aconseguirem posant a l'àrea de muntatge la peça, arrossegant-la.



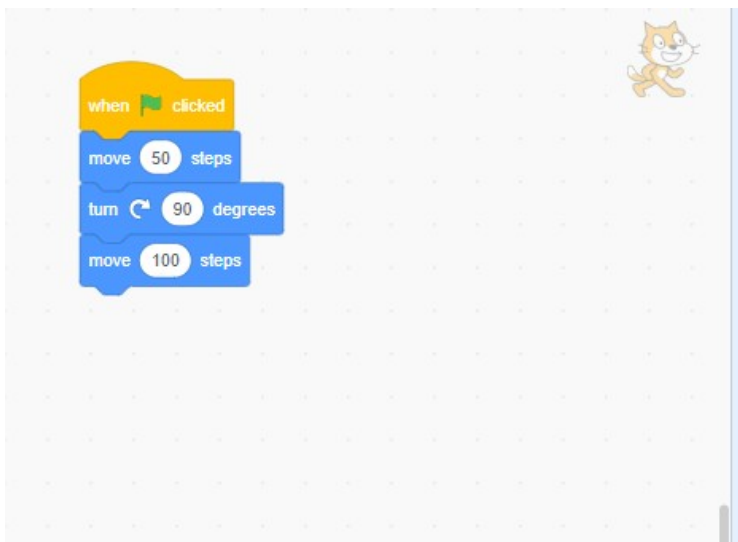
Després, hem d'indicar al programa que volem que el nostre personatge es desplaci cap a la dreta, això ho aconseguirem mitjançant la peça “**move steps**” a la secció “**motion**”.



A la peça que hem situat a l'àrea de muntatge, podem especificar un nombre concret de passos (*steps*). Si posem un número molt alt, potser hem de dir adéu al nostre estimat gat.

Per últim, per fer que el nostre estimat gat baixi, haurem de, en primer lloc, fer servir la peça *turn* (girar, en aquest cas, cap a la dreta) indicant 90 graus (*degrees*) i, per últim, tornar a fer servir una peça moure (*move*) perquè es mogui de forma considerable cap avall.

Executem mitjançant el *click* de la bandera verda. El programa hauria de quedar tal que així, i el gat (o la figura que haguem triat) hauria de moure's de la forma que hem programat.

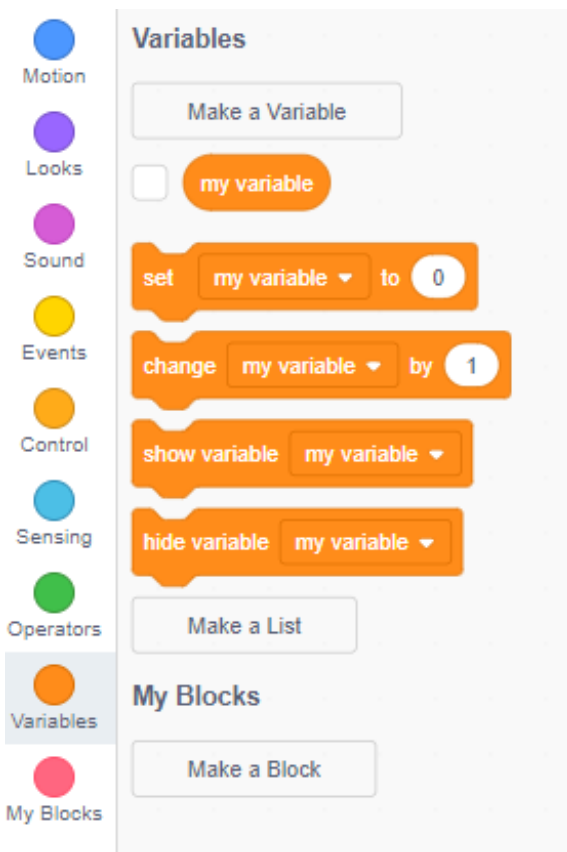


Imagineu que podem parametritzar la quantitat de graus o passos mitjançant **variables**, que puguem operar aquestes variables o afegir lògiques una mica més complexes mitjançant **condicionals** o **bucles**. Serà llavors quan us comenceu a adonar de la quantitat de coses que podreu arribar a fer amb només aquestes eines bàsiques.

Que es prepari el nostre estimat gat :)

5. Variables

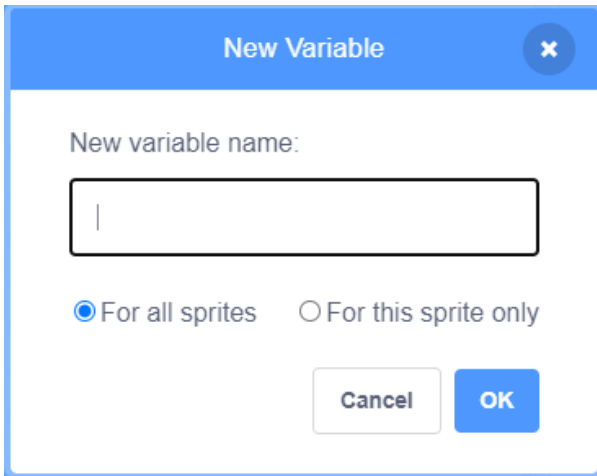
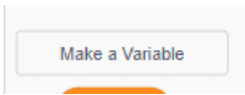
Declarar variables és força senzill en Scratch. Només heu de *clickar* a la secció “Variables” del menú esquerre i veureu tot el relacionat amb aquest tema.



D'aquí el que ens interessa per ara realment són les opcions:

- **Make a variable:** Serà aquí on podrem crear una variable i donar-li un determinat valor. A Scratch, en principi juguem amb valors numèrics, però recordeu que els llenguatges de programació que veurem tindrem altres tipus de valors bàsics, com per exemple booleans.
- **My variable:** En aquesta part sortiran llistades totes les variables creades. Si fem *click* al *checkbox* que acompanya a cadascuna de les variables, aquesta es mostrarà a la finestra d'execució o no en funció de l'estat d'aquest *checkbox*.
- **Set:** Assigna un determinat valor a una variable concreta.
- **Change:** Incrementa o decrementa el valor d'una variable prèviament inicialitzada.

5.1 Make a variable



New Variable

New variable name:

☒ For all sprites ☐ For this sprite only

Cancel OK

Indiquem el nom de la variable.

Les opcions de sota son per fer referència a si volem que afecti a un *sprite* (dibuix) en concret o a tots.

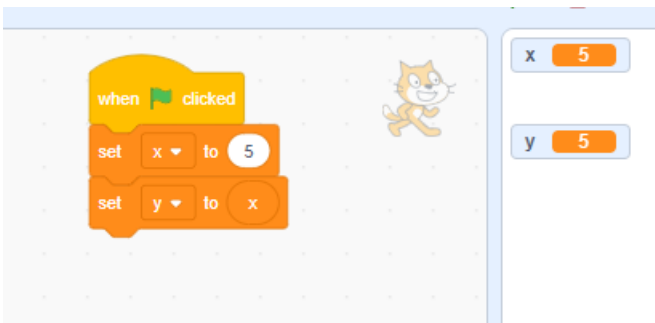
5.2 Select variable



Si fem *click* al *checkbox* d'una variable al menú de variables, aquesta es mostrarà junt amb el seu valor actual a la zona d'output.

5.3 Set variable

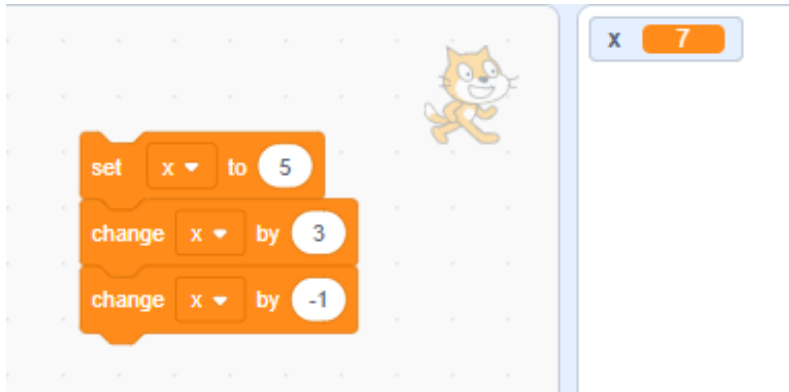
Podem assignar un valor a una variable amb la peça *set*, sigui indicant un número o una altra variable.



A la variable x li assignem el valor 5, Mentre que a la variable y li assignem el valor de dins de x.

5.4 Change variable

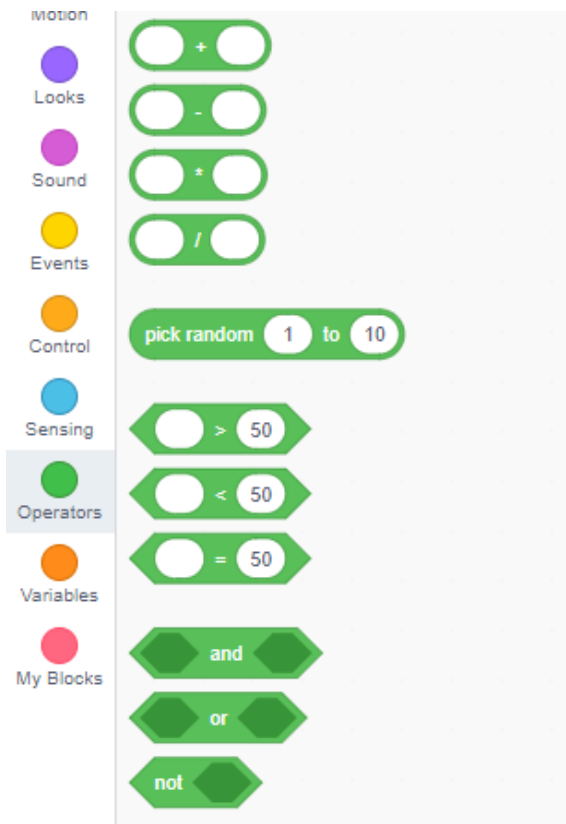
Podem incrementar/decrementar valors de variables amb l'opció/peça "change".



- 1 - A la variable x se li assigna el valor 5
- 2 - S'incrementa el valor de x en 3
- 3 - Es decrementa el valor de x en 1

6. Operadors

Recordeu que amb les variables podem fer diferents operacions i Scratch això també ens ho facilita força en la secció "Operators".



Treballarem amb els operadors aritmètics bàsics (suma, resta...), però fixeu-vos també en les expressions que retornen valors booleans (*true/false*) com per exemple:

> 50
 < 50
 $= 50$

Perquè les farem servir més endavant :)



Aquest exemple és força complet. Fixeu-vos com, encaixant peça darrera peça tenim un programa que s'inicia al *clickar* a la bandera verda i:

1- Inicialitza la variable x amb valor 5.

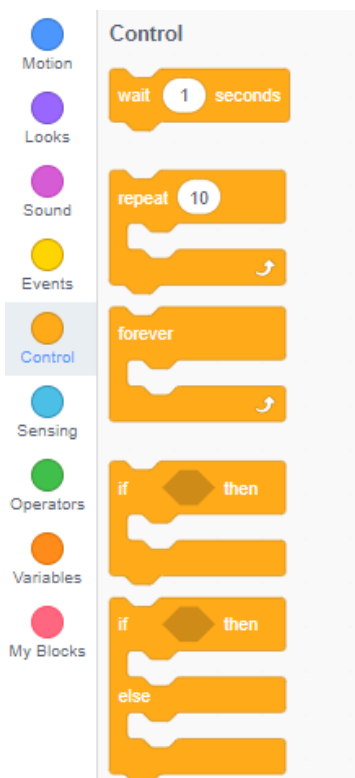
2- Inicialitza la variable y amb valor 3.

3- Assigna a la variable “resultat_suma” el valor de la suma de les variables x i y. Pareu atenció a com, encaixant les peces de diferents tipus, obtenim una expressió més complexa.

4- Igual que al pas 3, però amb la resta.

7. Sentències condicionals

Tal com vam explicar al document de conceptes bàsics de la programació, sovint, el *software* que desenvolupem haurà de realitzar preguntes per decidir uns o altres fluxos d'execució/conjunts d'instruccions. Scratch ens ho permet fàcil amb les opcions que ens mostra a la secció “Control”.



Les peces que farem servir ara son les peces if i if/else.



En aquest exemple recuperem els operadors que retornen valors booleans (en aquest cas, l'operador ">")

- Assignem valors a les variables x i y.
- En funció de si el valor de x és més gran que el de y s'executa una sentència o altra.

Observeu com aquí juguem amb la peça verda per definir una condició concreta.

8. Sentències de repetició

Hem parlat també de la necessitat que tenim a vegades que es repeteixin algunes instruccions de càlcul. A Scratch tenim les peces corresponents a aquesta lògica, són a la mateixa secció "**Control**", i són, en concret la peça "repeat" i la peça "repeat until".

Posem com a exemple un algoritme que calculi la potència d'un número.



Aquí tindríem un altre exemple on la condició que avalua el bucle és **booleana**, i on hem de ser sempre especialment curosos/es en què en algun moment es trenqui el bucle per no caure en bucle infinit.



La principal variació aquí és que a cada iteració del bucle, es comprova la condició “exponent = 0”, per tant, hem de garantir que això passa (en aquest cas mitjançant la instrucció que resta 1) per tal d’evitar bucles infinits.

9. Conclusions

Fins aquí aquest petit tutorial que aterra conceptes bàsics de programació en l’eina Scratch. Com heu pogut observar, hi ha moltes més opcions amb què jugar, no deixeu de fer-ho. Aquí només ens hem centrat en les que estan més directament relacionades al que tractareu en un principi al curs de “Fonaments de la programació” de tal manera que, quan hagueu de traslladar aquestes lògiques al codi font, imaginar-ho us resulti una mica més senzill.